Matthew Sanchez

CSC5-47982

Project 2

Mastermind

Directions

Mastermind is ultimate game of logic. The game starts with a computer generating four random colors. Begin by trying to guess the correct pattern. After each guess you will be hinted on the number of colors that were correct and in the right position. You will also be hinted on the number of colors that you entered that were correct but in the wrong spot. In this game you will be given 10 tries to get the correct sequence. Good luck!

Libraries: iostream, cstdlib, ctime

Lines of Code: 348

Total Variables: 48

Total Functions: 19 (pass by reference, and pass by value)

Total number of loops: 13

Total number of function calls: 36

Total number of decisions: 39

Pseudocode

Main

1. Declare Variables
2. Create 4 variables to convert random colors input to integer
3. Create an array variable for the random numbers/colors generated
4. Create an array variable for the users guesses
5. Create a counter variables for user guesses
6. Create a variable that inputs users decision to play again
7. Create a variable to count number of user guesses and initialize it to zero
8. Create 5 variables to count each color in the correct code
9. Create 5 variables to count each color in users guess code
10. Create 5 variables to store the amount kept when comparing the color counts
11. Create another variable to hold the sum of those values referred to in number 10 above.
12. Call function to display instructions
13. Call RNG function to generate the correct sequence
14. Range the random numbers 1-5
15. Assign each random number to the array “Correct”
16. Convert each random number to a character labeled as a color with the function
17. Call function call to output for user to start guessing
18. Input each user guesses into array “Guess”
19. Call function determine how many are correct
20. Call function to determine how many are the right color but in wrong position
21. Break out if user wins
22. Create a counter for user guesses
23. Add 1 to counter after each user guess input
24. Limit user guesses to 10
25. Determine if user wins
26. If user loses output correct sequence
27. Prompt user to play again
28. Exit Stage Right

Functions

1. Create a function to display instructions
2. Create a function to get user input
3. Prompt user to enter guesses and convert lower case entries to uppercase
4. Create a function with a switch menu to convert inputs to characters

a. Return each character

1. Create a function to check which guesses are correct color and in correct position
2. Create another function to determine if color is correct but in wrong position

Project Summary

The project took about a week and a half and was rather difficult for me. From my previous knowledge of taking this course last semester, I assumed using a couple of arrays would be best for my code even though we haven’t covered them yet. I also implemented everything we have covered so far.

* Data types int, bool ,char etc
* Void functions and functions that return data
* Pass by reference and value functions
* Arrays were also implemented
* Static casting to other data types
* ASCII table was implemented to convert uppercase to lowercase inputs
* Random function
* Time function
* If statements and if else statements
* Counter controlled loops
* Do while loops
* While loops
* Switch statement
* Modulus function

The code and flow chart were submitted separately inside the folder.